Simultaneous Localization and Mapping with Learned Object Recognition and Semantic Data Association

John G. Rogers III, Alexander J. B. Trevor, Carlos Nieto-Granda and Henrik I. Christensen

Abstract— **Complex and structured landmarks like objects have many advantages over low-level image features for semantic mapping. Low level features such as image corners suffer from occlusion boundaries, ambiguous data association, imaging artifacts, and viewpoint dependance. Artificial landmarks are an unsatisfactory alternative because they must be placed in the environment solely for the robot's benefit. Human environments contain many objects which can serve as suitable landmarks for robot navigation such as signs, objects, and furniture. Maps based on high level features which are identified by a learned classifier could better inform tasks such as semantic mapping and mobile manipulation. In this paper we present a technique for recognizing door signs using a learned classifier as one example of this approach, and demonstrate their use in a graphical SLAM framework with data association provided by reasoning about the semantic meaning of the sign.**

I. INTRODUCTION

Bridging the gap between mobile robots operating in the factory and operating in everyday environments requires the development of SLAM techniques and semantic reasoning. The inclusion of object-level landmarks in maps facilitate tasks such as object retrieval and more generalized human robot interaction dialog.

Complex and structured landmarks such as objects have many advantages over low-level image features for semantic mapping. Low-level features suffer from viewpoint dependent imaging conditions such as boundary occlusion (where the feature is on a boundary and will appear different in subsequent frames due to motion parallax), insufficient invariance to robot motion, and specular reflections.

Data association is a problem for most SLAM algorithms operating in unstructured environments. Low-level features make use of validation gates and joint compatibility to mitigate this problem; however, the use of higher level features reduces the significance of this problem, since each landmark might have uniquely identifiable characteristics. Signs, for example, often contain text which can be read by the robot to give a unique string which could be used as an unambiguous data association cue.

Semantic mapping also offers an advantage for robots to understand task assignments given to them by human users. Non-technical users will prefer human terms for objects and locations when assigning tasks to robots instead of whatever indices or coordinates the robot uses to represent them in its memory. A text string which could be read from a sign, such as "Room 213" provides semantic information both as a label, associating "213" with the present region and denoting the place as a "room".

In this paper, we present a method for using a learned object classifier in a SLAM context to provide measurements suitable for mapping. To demonstrate this, we present a classifier for recognizing door signs and a data association technique based on reading text in a graphical SLAM framework for an office environment.

Related work will be presented in Section II. The specific algorithms and techniques used in this paper will be presented in Section III and Section IV. The experimental procedure will be outlined in Section V. Results will be shown in Section VI along with some additional techniques which were developed to improve results. Conclusions will be presented along with our future plans in Section VII.

II. RELATED WORK

The Simultaneous Localization and Mapping problem (SLAM) has been addressed by the robotics community since the late 80's. One of the first working solutions to the SLAM problem was reported by Smith and Cheeseman in [17] which expanded the Extended Kalman Filter (EKF) to jointly represent the landmark positions along with the robot pose. A more complete treatment of the history of the SLAM problem can be found in the summary papers by Durrant-Whyte and Bailey [7], [1].

More recently, Dellaert determined that the graph SLAM problem could be addressed through the application of sparse linear algebra techniques. This technique is known as the Square Root SAM algorithm which was first reported in [5]. This implementation used sparse Cholesky factorization to efficiently optimize the robot trajectory and landmark locations. Dellaert has developed the *GTSAM* library which uses factor graphs to represent the landmark measurements and robot odometry. We use this library for our map optimization.

Castle *et. al.* has incorporated known planar objects in [3] as part of visual SLAM. This technique extracts SIFT features [13] from the image and periodically finds inliers to a homography from a canonical view of the known objects. Our work differs from this technique in that the object recognition module is not finding matches to a small set of known objects but is based on classification of objects which may not have been seen before.

Recognition and reading of door signs was proposed by Tomono *et. al.* in [18]. This work recognized and read specific door signs in a building and estimated their relative pose with respect to the robot. More recently Tomono *et.*

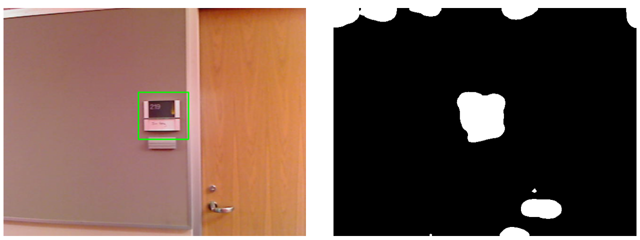
J. Rogers, A. Trevor, and C. Nieto-Granda are Ph. D. students at Georgia Tech College of Computing {jgrogers, atrevor, carlos.nieto}@gatech.edu

H. Christensen is the Kuka Chair of Robotics at Georgia Tech College of Computing hic@cc.gatech.edu

al. have developed an object recognition scheme which they used with a mapper to build object maps [19]. In contrast to this work, our approach uses a machine learning technique for recognition which could be extended to other types of objects.

III. DOOR SIGN DETECTION

To demonstrate our technique for mapping using learned object classifiers, we selected door signs as these are landmarks that frequently appear in human environments. They provide strong cues for navigation to humans, and can do so for robots as well. We have trained a classifier on door signs with widely varying appearances across several buildings. A description of the classifier's training and use is provided here to better demonstrate our method.



(a) Example image seen by robot (b) Saliency mask

Fig. 1. An image of a door sign seen by the robot is shown in figure 1(a) and the resulting saliency mask is shown in figure 1(b)

To enable efficient online operation, a method for selecting image regions as candidate door signs for our classifier was developed. The first step is to identify regions of the image which are visually distinctive. The spectral residual saliency technique of Hou and Zhang [12] is used due to its straightforward implementation and acceptable performance. A typical saliency result image is shown in Figure 1(b). The saliency image is analyzed to find blobs which are of appropriate size to be candidate sign regions. Occasionally, the saliency technique fails to isolate the sign, when the scene is too complex. For this reason, we have also added a series of fixed regions which cover the image frame at different sizes. This provides more chances to find the door signs when the saliency technique fails.

The door sign detection module uses the Histogram of Oriented Gradients feature for recognition [4]. Candidate door sign regions extracted using our saliency technique were hand-labeled as signs or not signs, and rescaled to a square 16x16 pixel size. A HOG feature was extracted from this region. A Support Vector Machine (SVM) was then trained on these features.

A. Histogram of Oriented Gradients

The HOG feature represents patterns of image gradients. Door signs are designed to be visually distinctive from the background to make them apparent to humans who are using them for navigation. Well designed signs typically exhibit sharp contrast from the background which generate strong edges under gradient analysis. HOG features were used with SVMs by Dalal and Triggs for classification in [4] where they were shown to perform well at recognizing people in images.

HOG features are constructed from an image window by first normalizing image contrast over a pattern of overlapping sub-windows within the window. We have chosen to only use one contrast normalization window because we empirically determined that the door sign is small enough to not exhibit significant contrast variation. The image is convolved with a set of oriented edge filters. These edge filter responses are binned into histograms which are positioned in a regular grid covering the initial window.

We selected HOG feature parameters to maximize recognition performance using coordinate ascent. HOG feature parameters which can be adjusted include the number of bins (edge filter orientations), the size of the detection window, and the contrast normalization window selection. We chose 16x16 pixel window size and 4x4 histogram cells per window and 9 orientations through 3-fold hold-out cross validation experiments. We did not detect an advantage for varying the overlapping contrast normalization regions in our application.

The HOG feature implementation from OpenCV [2] was used for this module.

B. Support Vector Machines

SVMs are a discriminative classifier which finds the maximum margin decision boundary on a kernel function of the input feature. The result of this learning procedure is a sparse set of *support vectors* which are the only components from the training set needed to perform classification. These support vectors are the examples which are nearest to the decision boundary. Classification is performed by evaluating the kernel function between the HOG feature of a candidate image region and each of the support vectors according to equation 1. In this equation, $y(x)$ is the predicted label, n is the number of support vectors, ω_n is the weight of the n-th support vector, $k(\cdot,\cdot)$ is a kernel function and b is an offset.

$$y(x) = \sum_{n=1}^{N} \omega_n k(x, x_n) + b \tag{1}$$

A variety of different kernel functions were tried for this recognition task including polynomials of degree 5 to 12, linear kernels, and the Radial basis function kernel (RBF). Polynomial kernels performed well in our cross-validation tests; however, the SVMs using them exhibited inferior generalization performance compared to the SVMs using RBF kernels. The γ parameter of the RBF kernel was selected via coordinate ascent to be 1.0.

The SVMs were trained with manually labeled regions extracted from images with our saliency technique detailed above; positive examples were provided which contained the door signs and negative examples were given of various other structures such as doorknobs, fire extinguishers, posters, door jambs, and random image regions.

We used the OpenCV [2] implementation of SVMs for the classifier in this paper.

C. Optical Character Recognition

Door signs in our building contain a unique recognition cue – a number and/or name identifying the room beyond the adjacent door. We use this cue for data association by reading the sign using a request to the GoogleGoggles server. We had previously attempted to use the open source optical character recognition (OCR) software library called *Tesseract* [16], but we found that while it works very well on analyzing scanned printed black-on-white text, it is difficult to adapt to camera images of text on signs.

Despite the significant performance improvements achieved by leveraging the GoogleGoggles service, sometimes the text strings returned contain a few errors which must be handled before they can be used for data association. These errors typically arise from the service matching non-text graphics or borders to a similarly shaped character, as well as missing text due to over or under segmentation due to lighting variability. To cope with these minor mistakes in reading, we first split the number component and the text string. A match in the number component results in a positive match. If the number fails to match, then the text string component is stripped of all whitespace and converted to lower case. The longest common subsequence is extracted between the mapped sign and the current measurement. If this subsequence is more than about 60% of the length of the longest of the two strings, then the text string results in a positive match.

D. Training

We trained classifiers on datasets from two different buildings on the Georgia Institute of Technology campus: the College of Computing building and the Klaus building. These datasets contained 105 and 133 images respectively, and approximately 10 to 20 positive and negative example regions were manually specified in each image as the training examples. Several example images from our training set are shown in Figure 2. 3-fold cross validation was performed for each classifier, and the results are summarized in Table I. The resulting SVMs were also tested on separate test data sets both from the same buildings, shown in Table II. While the classifiers have a low true positive rate for styles of signs that they weren't trained on, the false positive rate also remains low. The performance is a good match for use in our SLAM application, because missing a few true positives is acceptable, but adding false positives is highly undesirable. Also, we believe that the classifier's performance on the style of signs it has been trained on is more important than generalization to unseen sign types, because it is not unreasonable to imagine a robot that requires some amount of training specific to its intended environment. For the mapping experiments in this paper, we trained the classifier on images of the same types of signs as the test set from another part of the building. To be used for mapping, a sign which is selected by this classifier must also contain text which is understood and returned by GoogleGoggles. This condition further reduces the false positive rate so that no false positives have been observed during our testing.

Dataset	Positive	Negative	Num Vectors
CoC	0.953	0.985	262
Klaus	0.774	.963	1195
CoC-Klaus	0.795	0.955	1495

TABLE I

DOOR SIGN CLASSIFIER SVM CROSS VALIDATION RESULTS.

	CoC		Klaus	
CoC	0.954	0.960	0.321	0.984
Klaus	0.22	0.939	0.915	0.969
CoC-Klaus	0.908	0.949	0.915	0.974

TABLE II

SVM CONFUSION MATRIX RESULTS ON TRAINED BUILDINGS. TRUE POSITIVE RATE IS LISTED ON THE LEFT, TRUE NEGATIVE RATE IS LISTED ON THE RIGHT.

IV. MAPPING

Our robot makes use of the Robot Operating System (ROS) developed by Willow Garage [15] for control of the flow of data. Our technique uses three new software modules: the *laser-line-extractor*, the *door-sign-detector*, and the *mapper*. These modules will be explained in the following subsections.

A. Laser-line-extractor

Walls are extracted from straight lines in the laser scan. In Figure 3, example data is shown with extracted walls overlaid. We use a RANSAC [8] technique to extract lines from the laser data. This technique was adapted from the comparative analysis paper by Nguyen *et.al.* [14].

Pairs of points are uniformly selected from the laser point cloud (the laser range data rendered into a set of 2D points). Laser range data is analyzed to find collinear points to this line. If there are gaps in the laser line, these are used to break up one line into multiple lines. Only lines which are longer than a certain threshold are passed to the mapper as measurements.

B. Door-sign-detector

The door-sign-detector module makes use of the classifier described in Section III to recognize door signs in images taken from the robot's camera. If an image region is classified as a sign by the SVM, then a query is made from this image region to the GoogleGoggles server. If GoogleGoggles is able to read any text on the sign, then it will be returned to us in a response packet. Detected signs with decoded text are then published as measurements that can be used by the mapper. The measurements consist of the pixel location in the image of the detected region's centroid, the image patch corresponding to the detected region, and the text string returned from GoogleGoggles.

C. Mapper

Our SLAM implementation makes use of the GTSAM library [6]. This library represents the graph SLAM problem with a *factor graph* which relates landmarks to robot poses

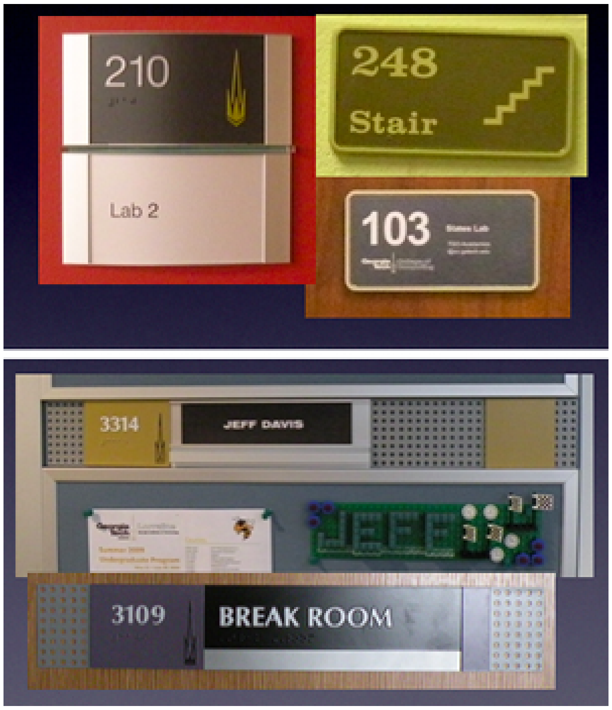


Fig. 2. Examples images of signs from our classifier's training set. Signs on the top are from the College of Computing dataset, and signs on the bottom are from the Klaus data set.

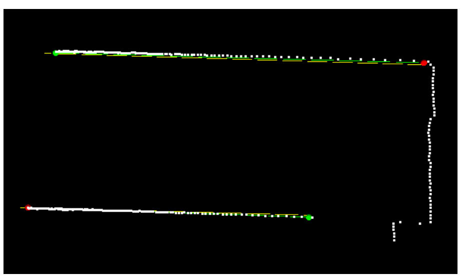


Fig. 3. Laser scanner data from the robot. The structure of the hallway is recognized by the laser line extractor as two walls on either side, shown with green lines. The wall at the end of the hall is too small in this view to be recognized as a line. Raw laser points are shown in white.

through factors. These factors are nonlinear measurements which are generated by the door sign detector and laser line extractor modules described above.

We have extended GTSAM with the Measurement space (M-space) feature representation was developed by Folkesson *et. al.* [10],[9], and [11]. The M-space feature representation allows us to use different types of features such as walls and points in a unified framework.

GTSAM builds a factor graph of nonlinear measurements. Each of these nonlinear measurements must support a *linearize* function which returns a *GaussianFactor* which is the linearization of its measurement function at the current configuration. The GaussianFactor takes an error vector and Jacobians relating the error vector to each of the variables involved in this factor. In our case, most M-space measurements involve the pose of the robot and the pose of a landmark such as a wall or a visual feature in the environment.

For the case of M-space walls, our linearization gives a measurement which is based on the line's range and orientation. The only errors in the line measurements are the angle of the line, and the perpendicular distance of the line to the origin. These parameters are $\eta = (\phi, \rho)$. The Jacobians which are needed by the linearization process to generate a GaussianFactor are $\frac{\delta\eta}{\delta x_r}$ and $\frac{\delta\eta}{\delta x_f}$ where x_r is the robot pose and x_f is the global landmark pose. In the M-space feature representation, it is desirable to use the chain rule to represent Jacobians in terms of smaller building blocks which can be re-used between different features. For example, the robot pose is common in all of these factors, so many of the terms will be the same between a wall and a point measurement. Details of this implementation, prior to the inclusion of the door sign features, can be found in previous work [20].

Originally, measurements of door sign features were implemented as projections of mapped features into image coordinates and direct comparison of pixel error from measured values. This approach proved unstable to initialize and did not work well when it was used to close extremely large loops with significant error. Measurements are now made on the 3D coordinates of the back-projected image location directly. Range is recovered by finding the laser beam from the head laser which is projects most closely to the image coordinates of the sign. This technique approximates the true range which we will eventually get from the use of a 3D camera like the Kinect. This factor also incorporates an additional variable which corresponds to the transformation between the robot base and the camera. By keeping track of the transformation when each measurement is taken, we are now able to move the camera on the pan-tilt unit during a data collection run.

To implement this factor in GTSAM, we must specify an error function and the error function's derivatives in terms of all of the variables which contribute to it. The error function is the difference in the 3D position of the predicted location of the sign from the measured value given by the recognition module.

V. EXPERIMENT

We performed a series of experiments to demonstrate the use of a learned classifier to generate landmark measurements in a SLAM context. We used the door sign classifier described in Section III to generate measurements from door signs in our office. The classifier used in this experiment was trained on images taken from a hand-held camera from a variety of different door signs on the second floor of our building. Multiple test runs consisting of different size loops were collected. Additionally, the training set was made from hand labeled and selected regions while the test run was made using the automatic saliency analysis and blob extraction and fixed sampling as explained in Section III. We also collected wall measurements from the laser scanner and used both feature types to generate maps.

The robot is shown in Figure 4. It is a Segway RMP-200 modified with external caster wheels. This modification

Fig. 4. The Segway RMP 200 with LMS 291 laser scanner for wall measurements and a Prosilica 650c camera with a Hokuyo UTM30 laser scanner on a PTU-46-70 pan-tilt unit. Four caster wheels were added for stability. The robot is shown in a position typical of reading a door sign.

Fig. 5. This sign is recognized and a measurement is made in the mapper. GoogleGoggles has read both the room number and the text, so this sign can be used for data association.

allows us to operate without using the balancing mode, which offers additional stability and safety. The robot makes use of a SICK LMS-291 laser scanner to collect measurements of walls and a Prosilica 650c camera with a Hokuyo UTM30 laser scanner mounted on a pan-tilt unit to collect images of door signs. The pan-tilt unit was controlled by the robot operator to point at the door signs during the test data collection. Camera images are collected automatically at a regular interval, not just when the camera is aimed at a door sign.

Currently, the robot is tele-operated in the environment while its sensor and odometry data are logged by ROS. The data file is processed offline by our system to construct the final map. This is done for convenience and repeatability, as well as to support our development. Our algorithms run in better than 2x real time with up to 353 poses and over 800 total measurements in the longest run. The transaction through GoogleGoggles server however does require about 4 seconds per frame, but this is only performed on image regions which are determined to be door signs. The mapper has been designed to operate asynchronously with measurement sources, so it can process messages arriving out of order from the recent past.

VI. RESULTS

A total of five test runs were performed, two runs at each of short and middle loop sizes, and one run at the large loop size. An example image is show in Figure 6 which illustrates the types of features which are mapped and how they are displayed in the maps. Some of the larger loops are big enough that it is hard to see the details in these images; please consult the accompanying video for additional details.

The short loop size is about 30 meters. In the runs at this loop size, the robot starts by proceeding down the west hallway and it is carefully driven and the camera is aimed at the door signs. The robot then drives through a cluttered laboratory space where few measurements can be made of the walls, resulting in significant pose error when the robot exits the lab. The robot is then driven back into the west hallway, but it doubles the hallway because of significant pose error, see Figure 7. In each of the test runs, the robot makes three successful sign matches and realigns the map, as shown in Figure 8.

The middle size loop run takes the robot first down the west hallway, and then onward into the half of the floor which is still under construction. Significant portions of this area contain clutter and are difficult to find wall segments large enough for mapping, resulting in some significant pose error in this portion. The robot then proceeds through the back hallways and around the loop for about 200 meters, where it re-enters the west hallway. The robot is now lost because the walls are not successfully matched due to significant pose error, see Figure 9. After door signs are matched, the robot is relocalized and the map is corrected in Figure 10.

The long run starts out the same as the middle length run but instead of proceeding back to the west hallway after exiting the back hallways and the construction area, the robot proceeds around the largest loop possible on our floor by driving down the east hallways and through the kitchen and atrium before returning to the west hallway. As with the other runs, the robot has become lost by the time it re-enters the west hallway and is unable to close the loop using only the wall features, see Figure 11. Once again, door signs are re-observed and the loop is closed, resulting in a useable map and a localized robot, see Figure 12.

VII. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

We have shown that a learned feature classifier can be used to detect objects which can then be mapped in SLAM. Specifically, we have demonstrated the use of an SVM to classify HOG features for mapping in graphical SLAM. The door signs selected as the object for detection possess se-

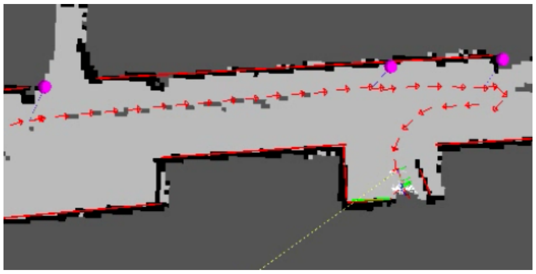


Fig. 6. A close-up of the west hallway. Robot poses are shown as red arrows. Wall features are shown as red lines. Door signs are shown as pink spheres. The occupancy grid is displayed only for clarity. This figure is best viewed in color.

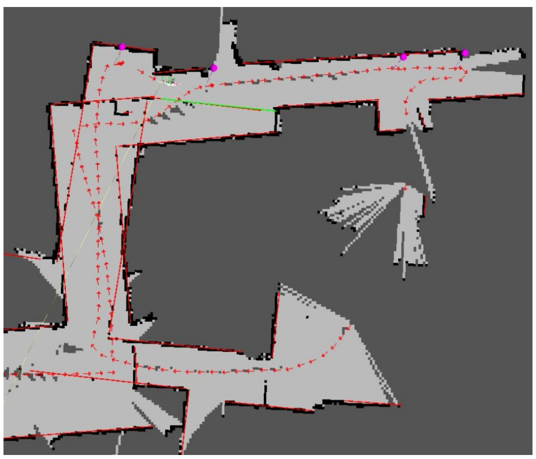


Fig. 7. A map through a cluttered lab space which has few line features for measurement, resulting in significant pose error and a failure to close the loop before re-measuring a door sign.

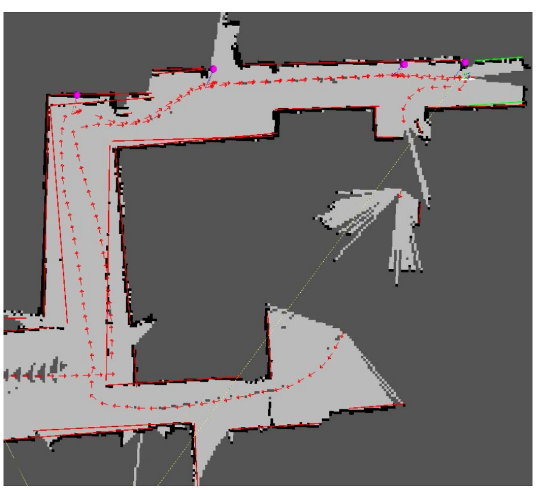


Fig. 8. A door sign is re-observed and the text is matched, resulting in a loop closure.

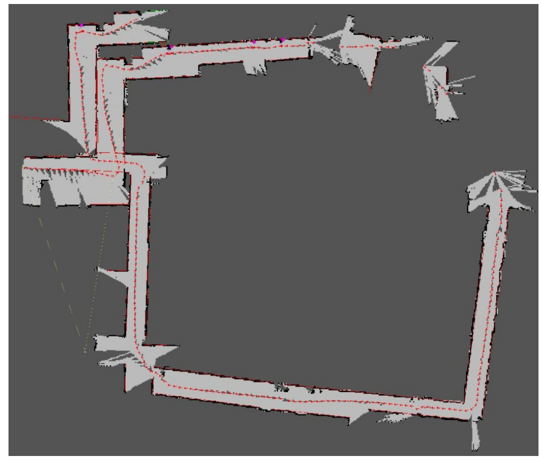


Fig. 9. A longer mapping run through the hallways in our building. The mapping run goes through a cluttered area under construction and gets lost, resulting in a several meter trajectory error, before a door sign is re-observed.

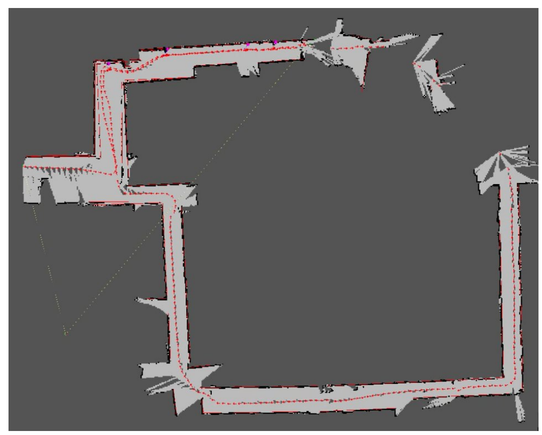


Fig. 10. Once a door sign is re-observed, the loop is closed and the map is corrected.

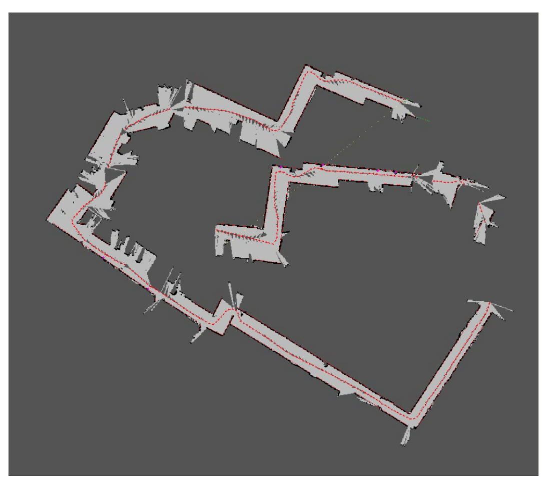


Fig. 11. The robot has completed the long loop around the building, but has not yet found a sign match to perform a loop closure. Note that there is significant error in this map when the door signs have not been re-observed.

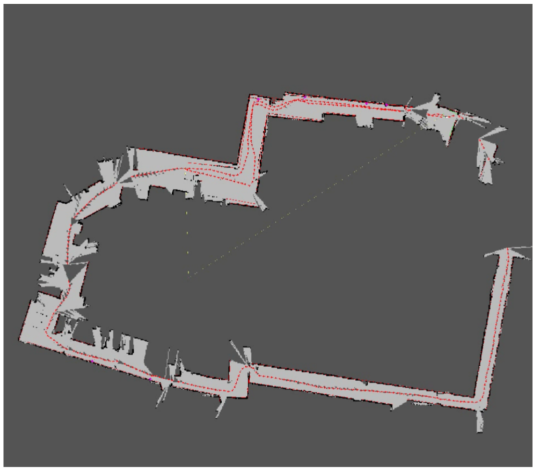


Fig. 12. The robot has proceeded further along and it has just re-observed a door sign from the first time around this loop. The map is corrected and the robot now knows where it is.

mantic information, the text describing the room beyond the door, which we use for data association to close large loops, relocalizing the robot when it was lost. The understanding of the semantic meaning of these sophisticated features enabled the robot to map in a large and complex environment where it would have otherwise become lost.

B. Future Works

The HOG feature classifier could potentially be used to recognize many different types of objects in the environment such as appliances, posters, and perhaps furniture. In addition to these *appearance based* techniques for object recognition, we could also consider using *model based* techniques which take into account the 3D model of the object being recognized. We wanted to leverage the generalization performance of appearance based techniques to recognize objects which have not been seen before. If we instead were able to establish correspondences between the image and a 3D object's pose through the use of a known CAD model, then a *pose measurement* could be given to the mapper instead of a less constraining *point measurement*. This technique could be used to recognize a small set of known objects but might not offer as much in terms of generalization to unseen examples of a class of objects, like door signs.

Signs are specific examples of objects which convey a great deal of information beyond just their position for localization. Most objects which would be of sufficient permanence and importance to be considered as valid for mapping would also probably be relevant to other tasks that the robot might be required to perform. An example would be finding appliances would inform the robot that it is in the kitchen, so it would know where to find other food preparation equipment. The door sign features with room numbers could also be used to identify locations in a semantic map for understanding human commands.

VIII. ACKNOWLEDGMENTS

This work was supported by grants from the ARL by the MAST CTA and the Boeing Corporation. Their support is gratefully acknowledged.

REFERENCES

[1] T. Bailey and H. Durrant-Whyte. Simultaneous localisation and mapping (SLAM): Part II state of the art. *Robotics and Automation Magazine*, September 2006.
[2] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.
[3] R. O. Castle, D. J. Gawley, G. Klein, and D.W. Murray. Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras. *International Conference on Robotics and Automation*, 2007.
[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 866, 2005.
[5] F. Dellaert. Square root SAM: Simultaneous localization and mapping via square root information smoothing. In *Robotics: Science and Systems*, 2005.
[6] F. Dellaert and M. Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 2006.
[7] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *Robotics and Automation Magazine*, June 2006.
[8] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *ACM New York, NY, USA*, 1981.
[9] J. Folkesson, P. Jensfelt, and H. Christensen. Vision SLAM in the measurement subspace. *International Conference on Robotics and Automation*, 2005.
[10] J. Folkesson, P. Jensfelt, and H.I. Christensen. Graphical SLAM using vision and the measurement subspace. *International Conference on Intelligent Robots and Systems*, 2005.
[11] J. Folkesson, P. Jensfelt, and H.I. Christensen. The M-space feature representation for SLAM. *IEEE Transactions on Robotics*, 2007.
[12] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. *CVPR*, 2007.
[13] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.
[14] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart. A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics. *International Conference on Intelligent Robots and Systems*, 2005.
[15] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully B. Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source robot operating system. In *International Conference on Robotics and Automation*, Open-Source Software workshop, 2009.
[16] R. Smith. An overview of the tesseract OCR engine. *Proceedings of the ICDAR*, 2007.
[17] R. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, Winter 1987.
[18] M. Tomono and S. Yuta. Mobile robot navigation in indoor environments using object and character recognition. In *International Conference on Robotics and Automation*, 2000.
[19] M. Tomono and S. Yuta. Object-based localization and mapping using loop constraints and geometric prior knowledge. In *International Conference on Robotics and Automation*, 2003.
[20] A. J. B. Trevor, J. G. Rogers III, C. Nieto-Granda, and H.I. Christensen. Applying domain knowledge to SLAM using virtual measurements. *International Conference on Robotics and Automation*, 2010.